DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

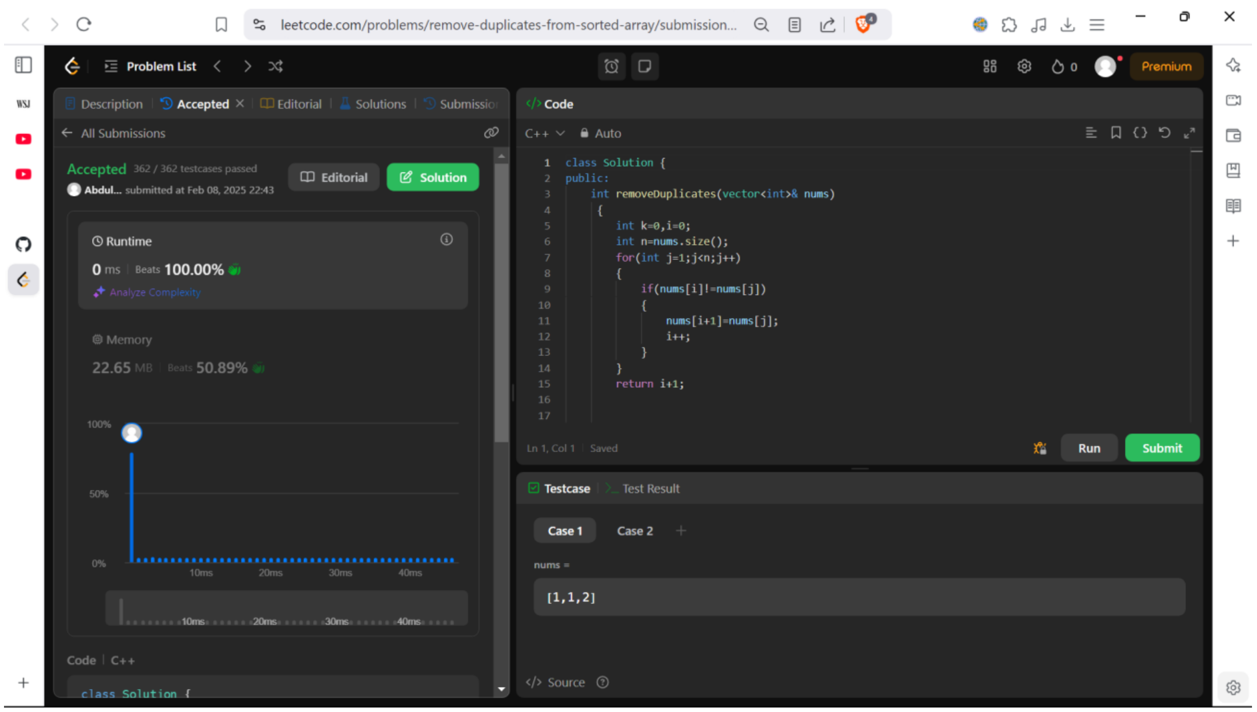
Assignment 1

Student Name: Abdul Kalam
UID: 22BCS14739
Branch: BE-CSE
Section/Group: IOT-601-B
Semester: 6th
Date of Performance: 22/07/24
Subject Name: Advanced Programming
Subject Code: 22CSP – 351

DAY-1:

1. Remove Duplicates From The Sorted Array-

```
int removeDuplicates(vector<int>& nums) {
      int k=0;int i=0;
      int size=nums.size();
      for(int j=1;j<size;j++){
         if(nums[i]!=nums[j]){
            nums[i+1]=nums[j];
            i++;
         }
      }
      return i+1;
   }
```


leetcode.com/problems/remove-duplicates-from-sorted-array/submission...

Accepted 362 / 362 testcases passed
Abdul... submitted at Feb 08, 2025 22:43

Runtime
0 ms | Beats 100.00%

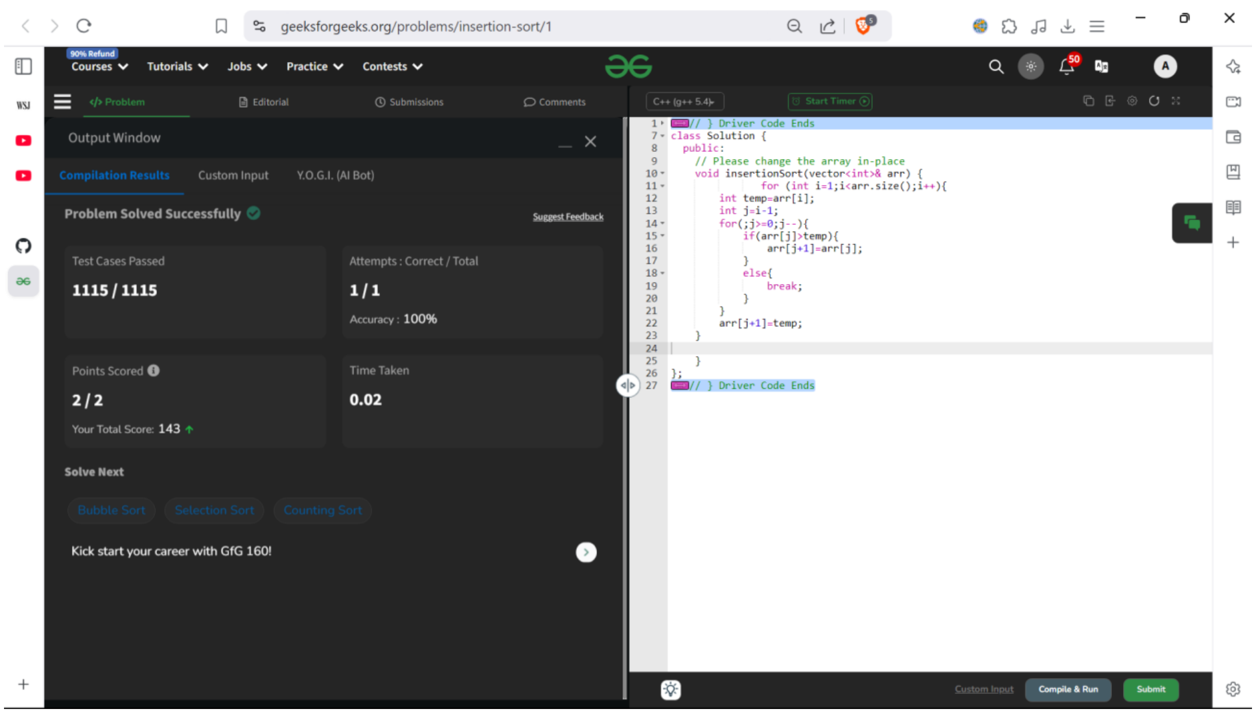
Memory
22.65 MB | Beats 50.89%

```
class Solution {
public:
    int removeDuplicates(vector<int>& nums)
     {
        int k=0,i=0;
        int n=nums.size();
        for(int j=1;j<n;j++)
        {
            if(nums[i]!=nums[j])
            {
                nums[i+1]=nums[j];
                i++;
            }
        }
        return i+1;

```

nums =
[1,1,2]

2. Implementing Insertion Sort-

```
void insertionSort(vector<int>& arr) {
        // code here
        for (int i=1;i<arr.size();i++){
        int temp=arr[i];
        int j=i-1;
        for(;j>=0;j--){
            if(arr[j]>temp){
                arr[j+1]=arr[j];
            }
            else{
                break;
            }
        }
        arr[j+1]=temp;
    }
    }
```



Problem Solved Successfully

Test Cases Passed
1115 / 1115

Attempts : Correct / Total
1 / 1
Accuracy : 100%

Points Scored
2 / 2
Your Total Score: 143

Time Taken
0.02

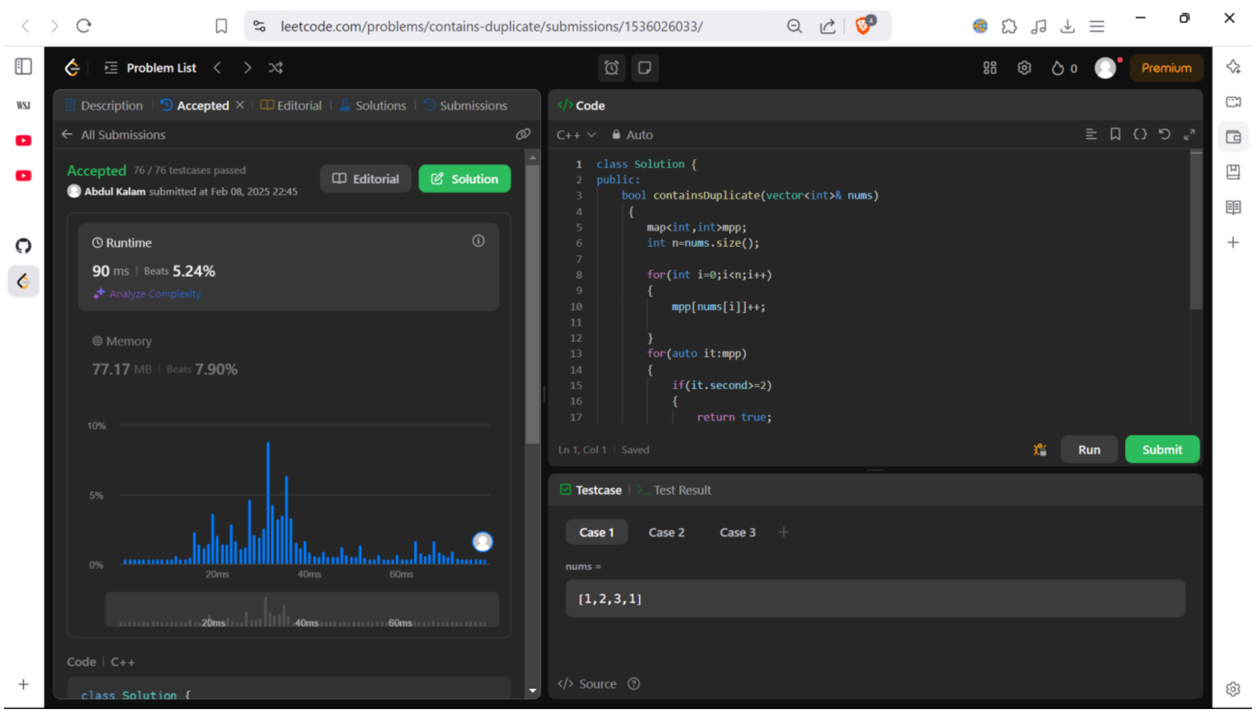
Solve Next
Bubble Sort Selection Sort Counting Sort

Kick start your career with GfG 160!

```
// } Driver Code Ends
class Solution {
  public:
    // Please change the array in-place
    void insertionSort(vector<int>& arr) {
                for (int i=1;i<arr.size();i++){
        int temp=arr[i];
        int j=i-1;
        for(;j>=0;j--){
            if(arr[j]>temp){
                arr[j+1]=arr[j];
            }
            else{
                break;
            }
        }
        arr[j+1]=temp;
    }

    }
};
// } Driver Code Ends
```

3. Contains Duplicate-

```
bool containsDuplicate(vector<int>& nums) {
       map<int,int>mpp;
        int n=nums.size();
        for(int i=0;i<n;i++)
        {
            mpp[nums[i]]++;
        }
        for(auto it:mpp)
        {
            if(it.second>=2)
            {
                return true;
            }
        }
        return false;
}
```



leetcode.com/problems/contains-duplicate/submissions/1536026033/

Accepted 76 / 76 testcases passed
Abdul Kalam submitted at Feb 08, 2025 22:45

Runtime
90 ms | Beats 5.24%

Memory
77.17 MB | Beats 7.90%

Code | C++

```
class Solution {
```

```
class Solution {
public:
    bool containsDuplicate(vector<int>& nums)
     {
        map<int,int>mpp;
        int n=nums.size();

        for(int i=0;i<n;i++)
        {
            mpp[nums[i]]++;

        }
        for(auto it:mpp)
        {
            if(it.second>=2)
            {
                return true;
```

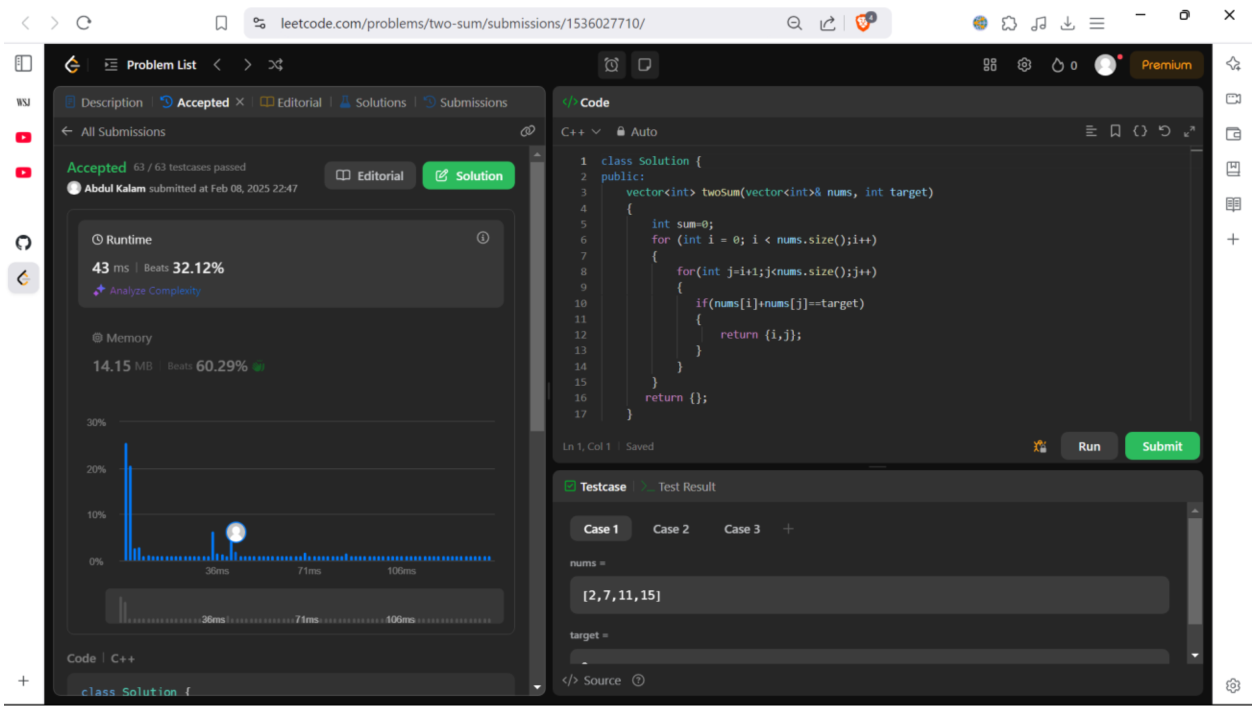
Ln 1, Col 1 | Saved

nums =

```
[1,2,3,1]
```

4. Two Sum –

```
vector<int> twoSum(vector<int>& nums, int target) {
         int sum=0;
        for (int i = 0; i < nums.size();i++)
        {
            for(int j=i+1;j<nums.size();j++)
            {
               if(nums[i]+nums[j]==target)
               {
                   return {i,j};
               }
            }
        }
       return {};
    }
```



Accepted 63 / 63 testcases passed
Abdul Kalam submitted at Feb 08, 2025 22:47

Runtime
43 ms | Beats 32.12%

Memory
14.15 MB | Beats 60.29%

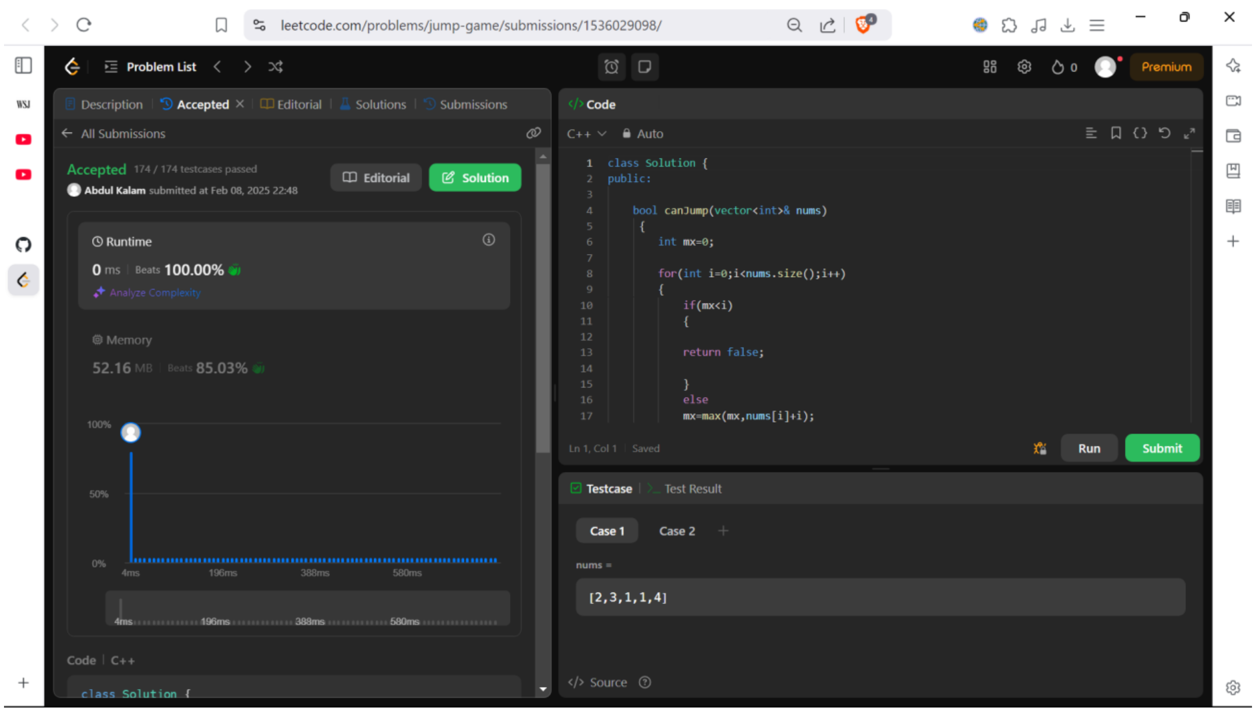
```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target)
    {
        int sum=0;
        for (int i = 0; i < nums.size();i++)
        {
            for(int j=i+1;j<nums.size();j++)
            {
               if(nums[i]+nums[j]==target)
               {
                   return {i,j};
               }
            }
        }
       return {};
    }
```

Ln 1, Col 1 | Saved

nums =
[2,7,11,15]
target =

5. Jump Game-

```
bool canJump(vector<int>& nums) {
int mx=0;
        for(int i=0;i<nums.size();i++)
        {
            if(mx<i)
            {
            return false;
            }
            else
            mx=max(mx,nums[i]+i);
        }
        return true;
    }
```



leetcode.com/problems/jump-game/submissions/1536029098/

Accepted 174 / 174 testcases passed
Abdul Kalam submitted at Feb 08, 2025 22:48

Runtime
0 ms | Beats 100.00%

Memory
52.16 MB | Beats 85.03%

Code | C++

```
class Solution {
public:

    bool canJump(vector<int>& nums)
     {
        int mx=0;

        for(int i=0;i<nums.size();i++)
        {
            if(mx<i)
            {

            return false;

            }
            else
            mx=max(mx,nums[i]+i);
```

Ln 1, Col 1 | Saved

Testcase | Test Result

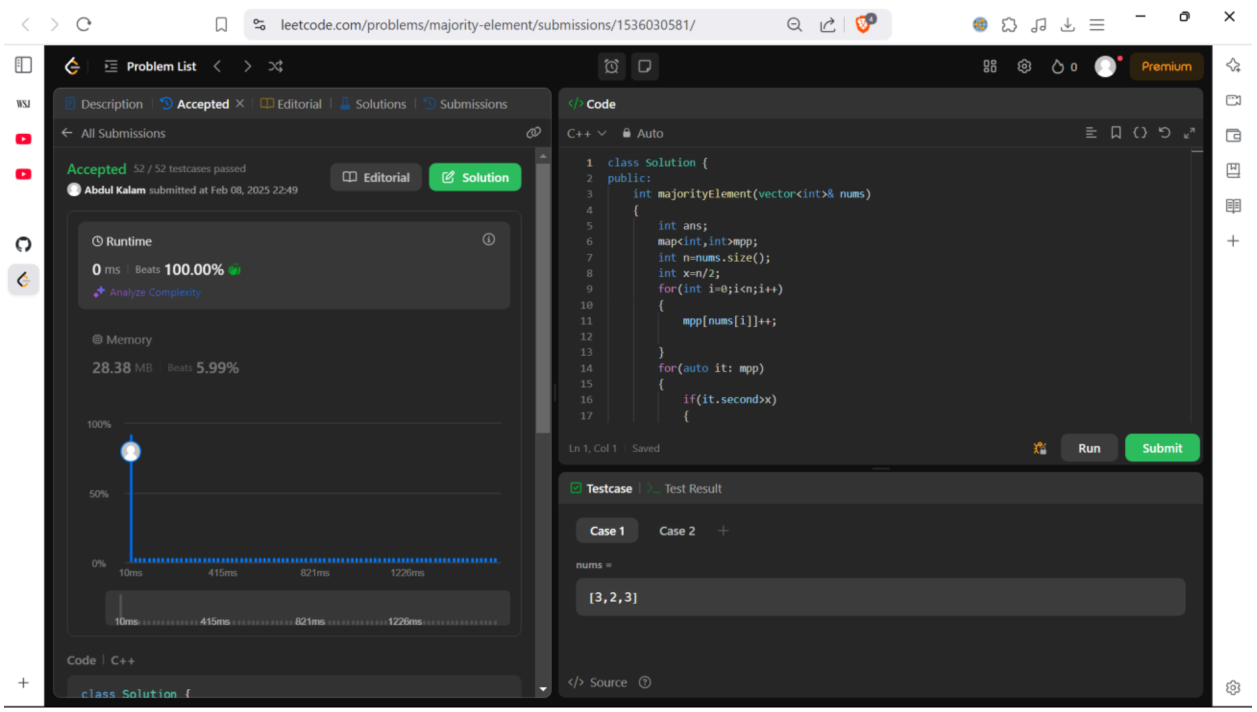
Case 1 | Case 2

nums =

```
[2,3,1,1,4]
```

6. Majority Element-

```
int majorityElement(vector<int>& nums) {
int ans;
      map<int,int>mpp;
      int n=nums.size();
      int x=n/2;
      for(int i=0;i<n;i++)
      {
         mpp[nums[i]]++;

      }
      for(auto it: mpp)
      {
         if(it.second>x)
         {
            ans=it.first;
         }
      }
      return ans;

   }
```


leetcode.com/problems/majority-element/submissions/1536030581/

Accepted 52 / 52 testcases passed
Abdul Kalam submitted at Feb 08, 2025 22:49

Runtime
0 ms | Beats 100.00%

Memory
28.38 MB | Beats 5.99%

Code | C++

C++ Auto

```
class Solution {
public:
    int majorityElement(vector<int>& nums)
    {
        int ans;
        map<int,int>mpp;
        int n=nums.size();
        int x=n/2;
        for(int i=0;i<n;i++)
        {
            mpp[nums[i]]++;

        }
        for(auto it: mpp)
        {
            if(it.second>x)
            {
```

Ln 1, Col 1 | Saved

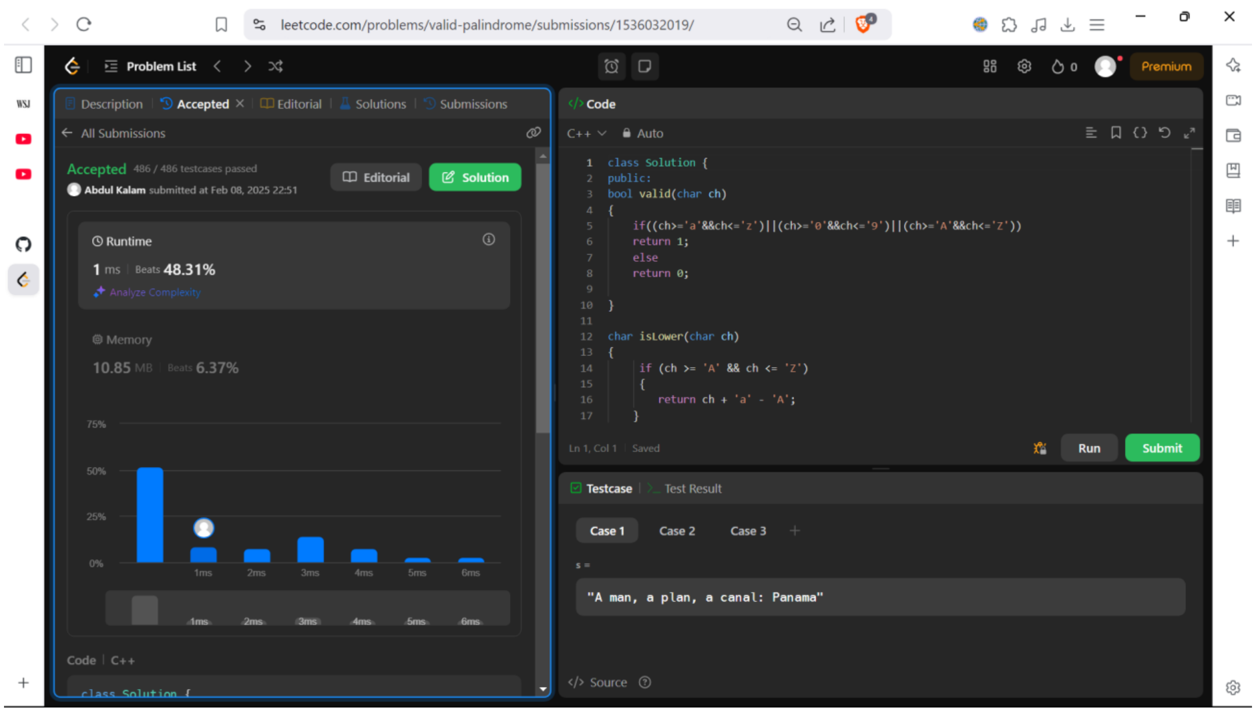
Case 1 Case 2

nums =

```
[3,2,3]
```

7. Valid Palindrome –

```
class Solution {
public:
    bool valid(char ch) {
        return (ch >= 'a' && ch <= 'z') || (ch >= '0' && ch <= '9') || (ch >= 'A' && ch <= 'Z');}
    char isLower(char ch) {
        return (ch >= 'A' && ch <= 'Z') ? (ch + 'a' - 'A') : ch;
    }
    bool Palin(string str) {
        int s = 0, e = str.length() - 1;
        while (s < e) {
            if (str[s++] != str[e--]) return false; }
        return true;  }
    bool isPalindrome(string s) {
        string temp = "";
        for (char ch : s) {
            if (valid(ch)) temp.push_back(isLower(ch));  }
        return Palin(temp);
    }
};
```


leetcode.com/problems/valid-palindrome/submissions/1536032019/

Accepted 486 / 486 testcases passed
Abdul Kalam submitted at Feb 08, 2025 22:51

Runtime
1 ms | Beats 48.31%

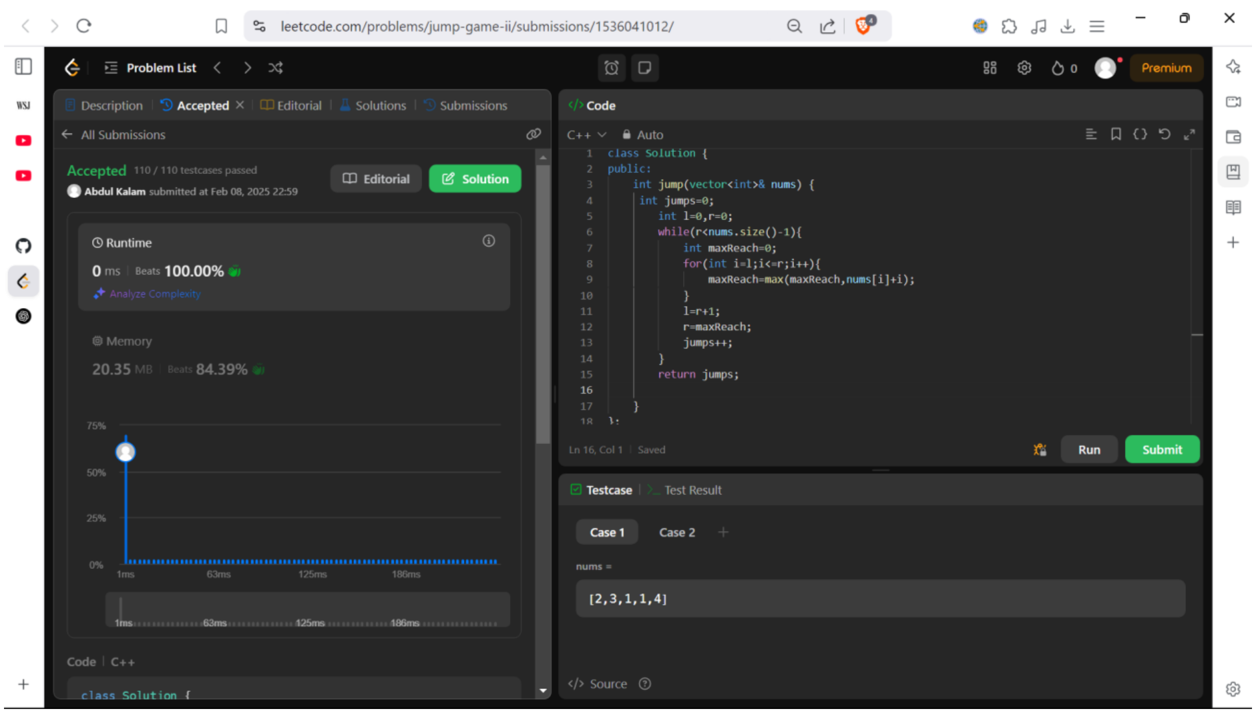
Memory
10.85 MB | Beats 6.37%

```
class Solution {
public:
bool valid(char ch)
{
    if((ch>='a'&&ch<='z')||(ch>='0'&&ch<='9')||(ch>='A'&&ch<='Z'))
    return 1;
    else
    return 0;

}

char isLower(char ch)
{
     if (ch >= 'A' && ch <= 'Z')
     {
        return ch + 'a' - 'A';
    }
```

s =
"A man, a plan, a canal: Panama"

8. Jump Game 2-

```
int jump(vector<int>& nums) {
        int jumps=0;
        int l=0,r=0;
        while(r<nums.size()-1){
            int maxReach=0;
            for(int i=l;i<=r;i++){
                maxReach=max(maxReach,nums[i]+i);
            }
            l=r+1;
            r=maxReach;
            jumps++;
        }
        return jumps;
    }
```


leetcode.com/problems/jump-game-ii/submissions/1536041012/

Accepted 110 / 110 testcases passed
Abdul Kalam submitted at Feb 08, 2025 22:59

Runtime
0 ms | Beats 100.00%

Memory
20.35 MB | Beats 84.39%

Code | C++

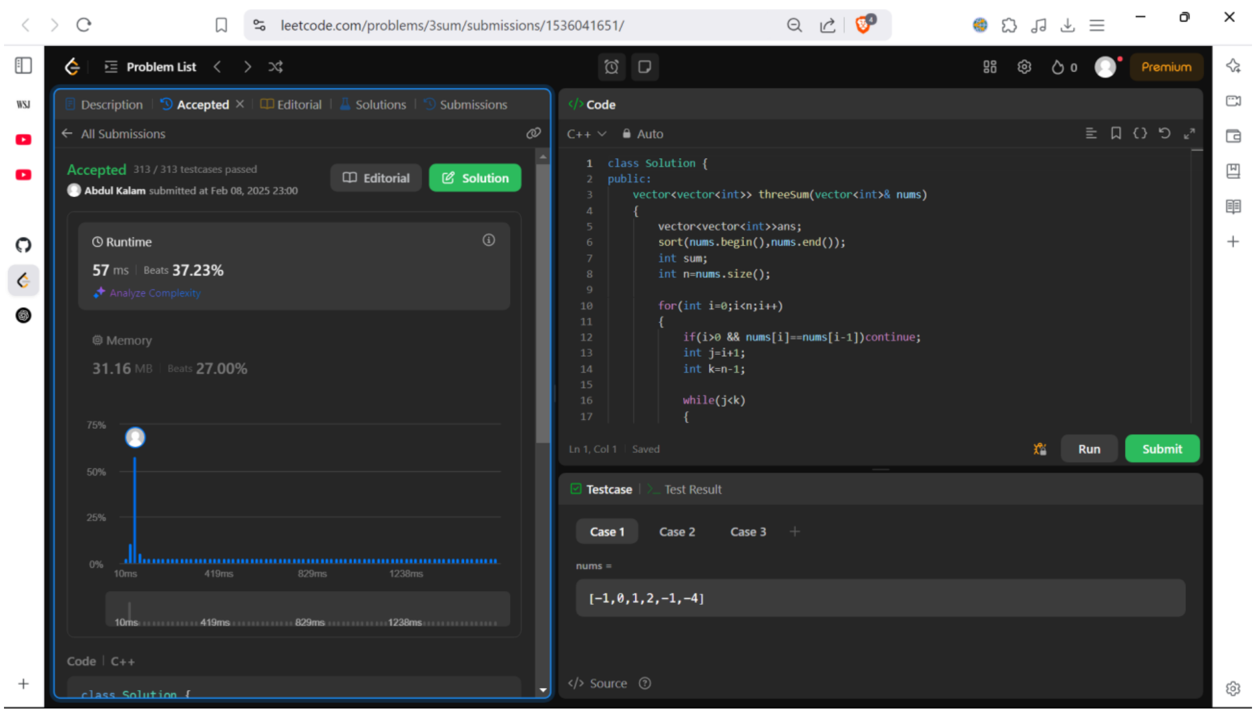
```
class Solution {
public:
    int jump(vector<int>& nums) {
     int jumps=0;
        int l=0,r=0;
        while(r<nums.size()-1){
            int maxReach=0;
            for(int i=l;i<=r;i++){
                maxReach=max(maxReach,nums[i]+i);
            }
            l=r+1;
            r=maxReach;
            jumps++;
        }
        return jumps;

    }
};
```

Ln 16, Col 1 | Saved

Testcase | Test Result
Case 1 Case 2
nums =
[2,3,1,1,4]

9. 3 Sum-

```
vector<vector<int>> threeSum(vector<int>& nums) {
vector<vector<int>>ans;
        sort(nums.begin(),nums.end());
        int sum;
        int n=nums.size();
        for(int i=0;i<n;i++)
        {
            if(i>0 && nums[i]==nums[i-1])continue;
            int j=i+1;
            int k=n-1;
            while(j<k)
            {
                sum=nums[i]+nums[j]+nums[k];
                if(sum<0)
                {
                    j++;
                }
               else if(sum>0)
                {
                    k--;
                }
                else
                {
                    vector<int> temp={nums[i],nums[j],nums[k]};
                    ans.push_back(temp);
                    j++;
                    k--;
                    while(j<k && nums[j]==nums[j-1])j++;
                    while(j<k && nums[k]==nums[k+1])k--;
                }
            }

        }
        return ans;
    }
```



Accepted 313 / 313 testcases passed
Abdul Kalam submitted at Feb 08, 2025 23:00

Runtime 57 ms | Beats 37.23%

Memory 31.16 MB | Beats 27.00%

Code | C++

```cpp
class Solution {
public:
    vector<vector<int>> threeSum(vector<int>& nums)
    {
        vector<vector<int>>ans;
        sort(nums.begin(),nums.end());
        int sum;
        int n=nums.size();

        for(int i=0;i<n;i++)
        {
            if(i>0 && nums[i]==nums[i-1])continue;
            int j=i+1;
            int k=n-1;

            while(j<k)
            {
```

Ln 1, Col 1 | Saved

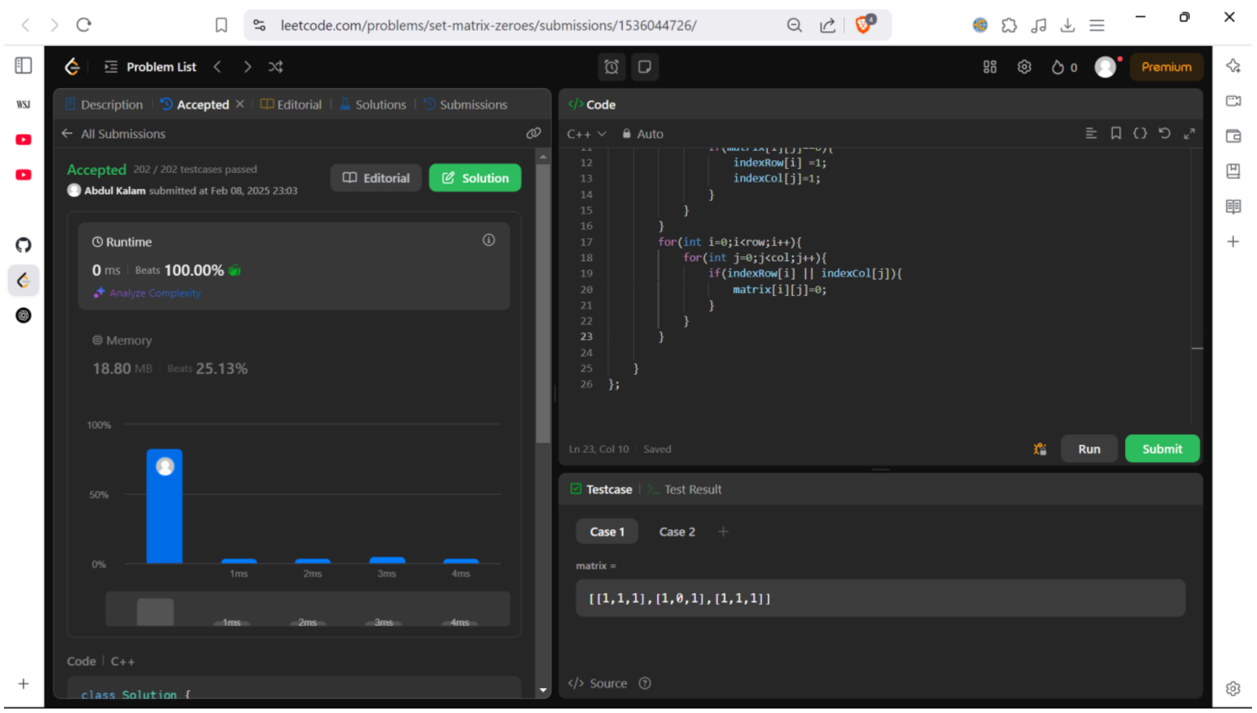
Testcase | Test Result

nums =

```
[-1,0,1,2,-1,-4]
```

10. Set Matrix Zeros-

```
void setZeroes(vector<vector<int>>& matrix) {
        int row=matrix.size();
        int col=matrix[0].size();

        vector<int> indexRow(row,0);
        vector<int> indexCol(col,0);

        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                if(matrix[i][j]==0){
                    indexRow[i] =1;
                    indexCol[j]=1;
                }
            }
        }

        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                if(indexRow[i] || indexCol[j]){
                    matrix[i][j]=0;
                }
            }
        }
    }
```



leetcode.com/problems/set-matrix-zeroes/submissions/1536044726/

Accepted 202 / 202 testcases passed
Abdul Kalam submitted at Feb 08, 2025 23:03

Runtime
0 ms | Beats 100.00%

Memory
18.80 MB | Beats 25.13%

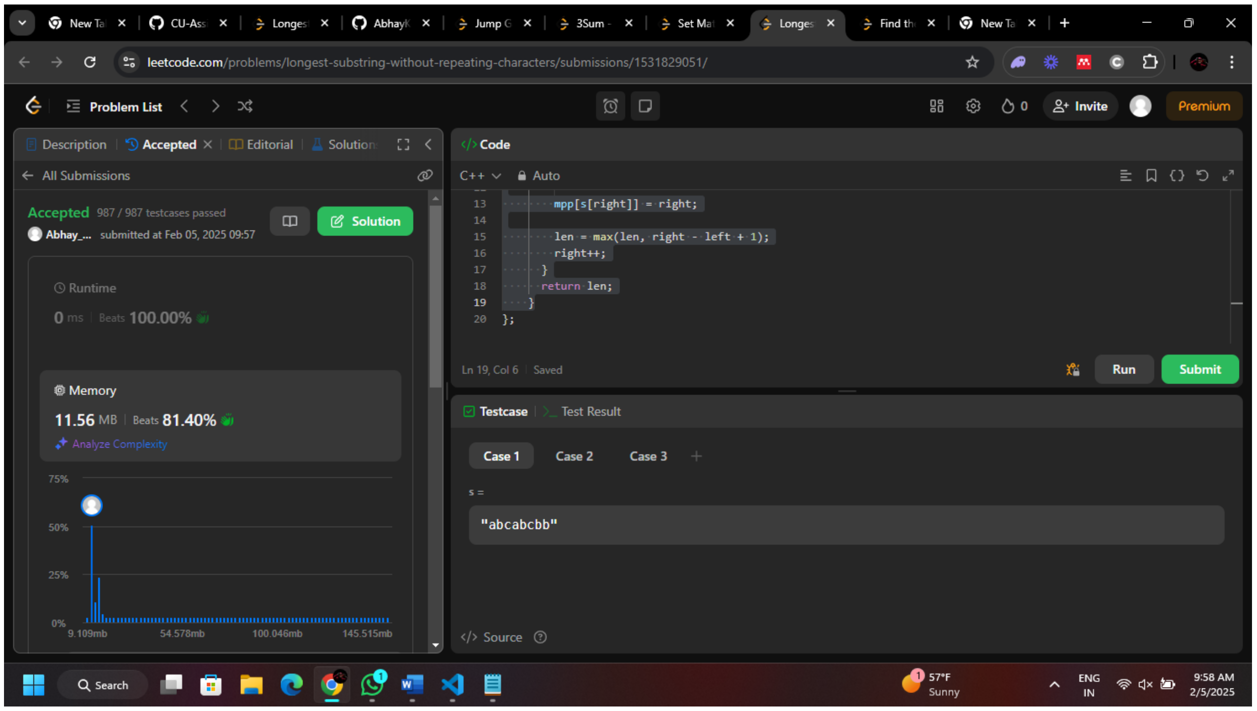
```
                if(matrix[i][j]==0){
                    indexRow[i] =1;
                    indexCol[j]=1;
                }
            }
        }
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                if(indexRow[i] || indexCol[j]){
                    matrix[i][j]=0;
                }
            }
        }

    }
};
```

Ln 23, Col 10 | Saved

matrix =
[[1,1,1],[1,0,1],[1,1,1]]

11. Longest Substring Without Repeating Characters-

```
int lengthOfLongestSubstring(string s) {
    vector < int > mpp(256, -1);

    int left = 0, right = 0;
    int n = s.size();
    int len = 0;
    while (right < n) {
      if (mpp[s[right]] != -1)
        left = max(mpp[s[right]] + 1, left);

      mpp[s[right]] = right;

      len = max(len, right - left + 1);
      right++;
    }
    return len;
  }
```


leetcode.com/problems/longest-substring-without-repeating-characters/submissions/1531829051/

Accepted 987 / 987 testcases passed
Abhay_... submitted at Feb 05, 2025 09:57

Runtime
0 ms | Beats 100.00%

Memory
11.56 MB | Beats 81.40%

```
        mpp[s[right]] = right;

        len = max(len, right - left + 1);
        right++;
    }
    return len;
  }
};
```

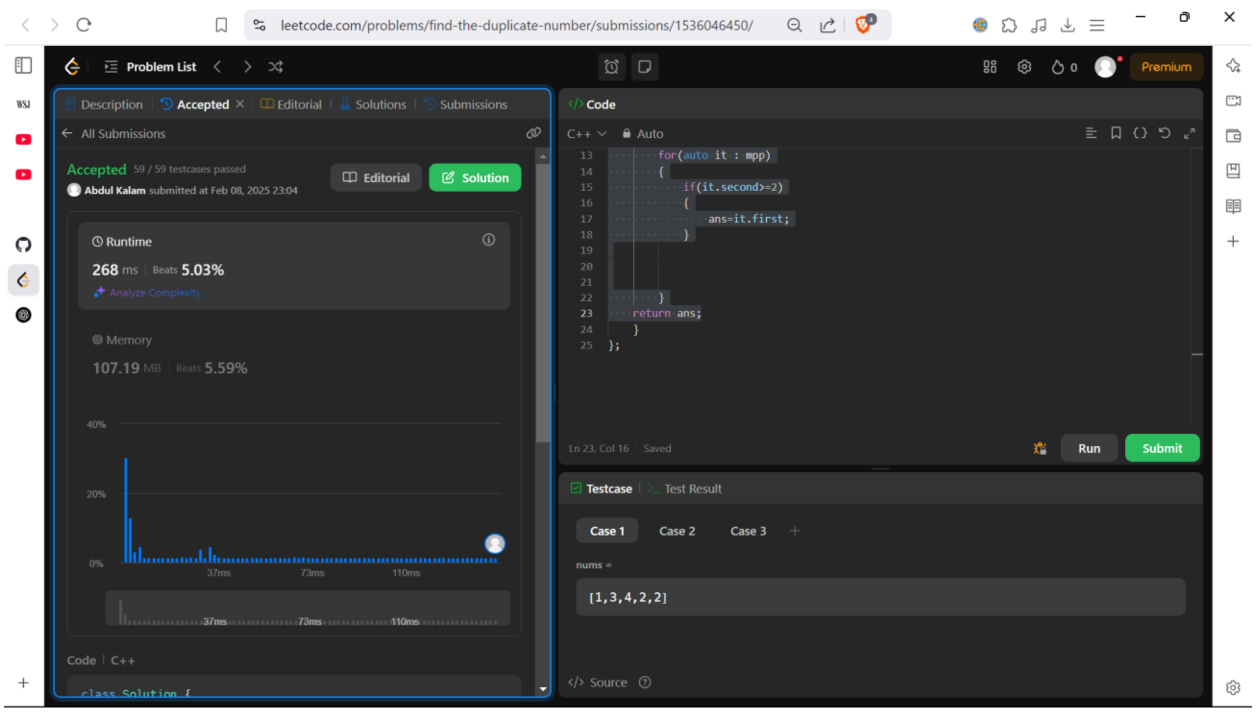
Ln 19, Col 6 | Saved

Testcase | Test Result
Case 1 Case 2 Case 3

s =
"abcabcbb"

12. Finding Duplicate Number-

```
int findDuplicate(vector<int>& nums) {
    int ans;
    map<int,int>mpp;
    int n=nums.size();
    for(int i=0;i<n;i++)
    {
        mpp[nums[i]]++;
    }
    for(auto it : mpp)
    {
        if(it.second>=2)
        {
            ans=it.first;
        }
    }
  return ans;
  }
```



Accepted 59 / 59 testcases passed
Abdul Kalam submitted at Feb 08, 2025 23:04

Runtime
268 ms | Beats 5.03%

Memory
107.19 MB | Beats 5.59%

```
        for(auto it : mpp)
        {
            if(it.second>=2)
            {
                ans=it.first;
            }



        }
    return ans;
    }
};
```

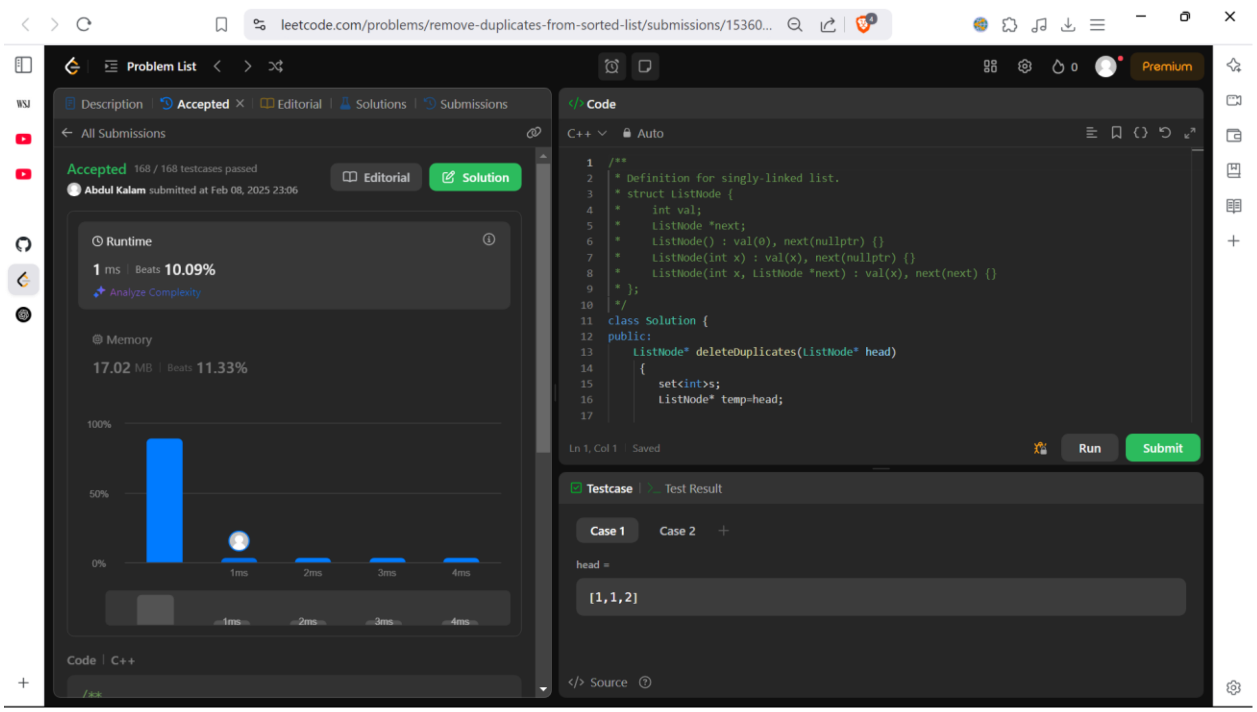
Ln 23, Col 16 | Saved

nums =
[1,3,4,2,2]

DAY-2:

1. Remove Duplicates From A Sorted List-

```
ListNode* deleteDuplicates(ListNode* head) {
        set<int>s;
        ListNode* temp=head;
        while(temp!=nullptr)
        {
            s.insert(temp->val);
            temp=temp->next;
        }
        ListNode* newHead=new ListNode(0);
        ListNode* current=newHead;
        for(auto it : s)
        {
            current->next=new ListNode(it);
            current=current->next;
        }
        return newHead->next;
    }
```



Accepted 168 / 168 testcases passed
Abdul Kalam submitted at Feb 08, 2025 23:06

Runtime
1 ms | Beats 10.09%

Memory
17.02 MB | Beats 11.33%

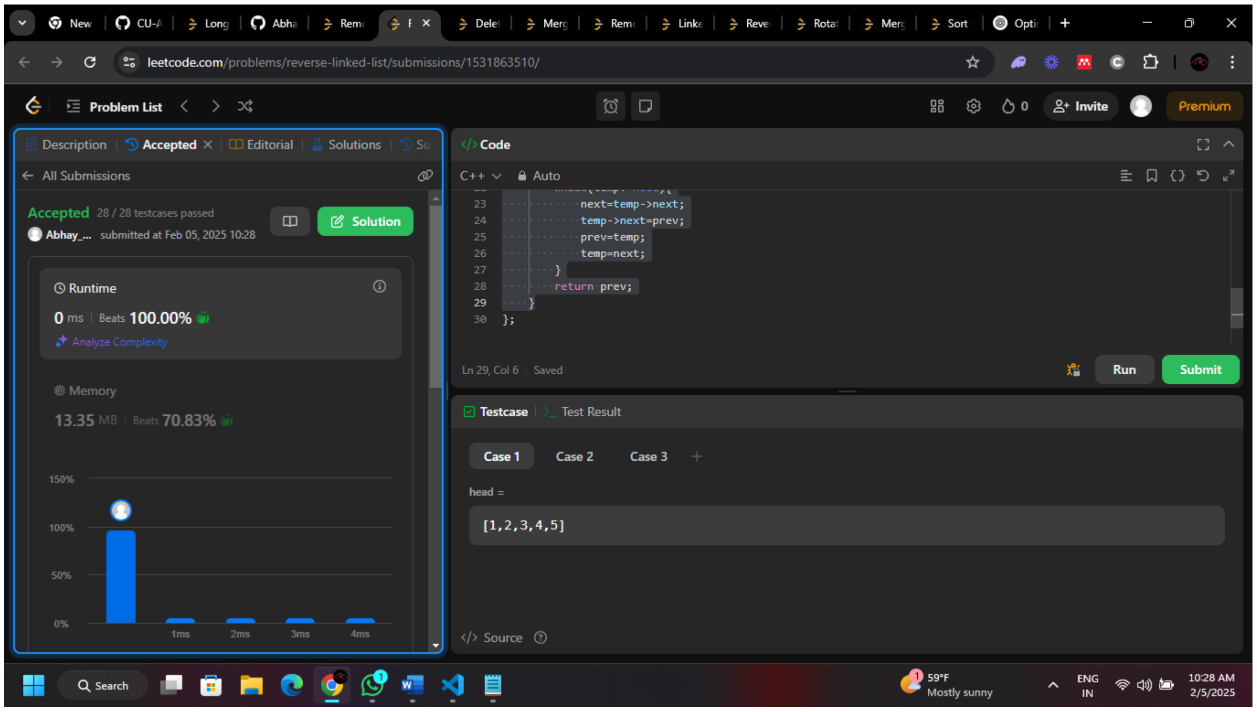
```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head)
    {
        set<int>s;
        ListNode* temp=head;

```

head =

```
[1,1,2]
```

2. Reverse A Linked List –

```
ListNode* reverseList(ListNode* head) {
        if(head==NULL || head->next==NULL){
            return head;
        }

        ListNode* prev=NULL;
        ListNode* temp=head;
        ListNode* next=NULL;

        while(temp!=NULL){
            next=temp->next;
            temp->next=prev;
            prev=temp;
            temp=next;
        }
        return prev;
    }
```



leetcode.com/problems/reverse-linked-list/submissions/1531863510/

Accepted 28 / 28 testcases passed
Abhay_... submitted at Feb 05, 2025 10:28

Runtime
0 ms | Beats 100.00%

Memory
13.35 MB | Beats 70.83%

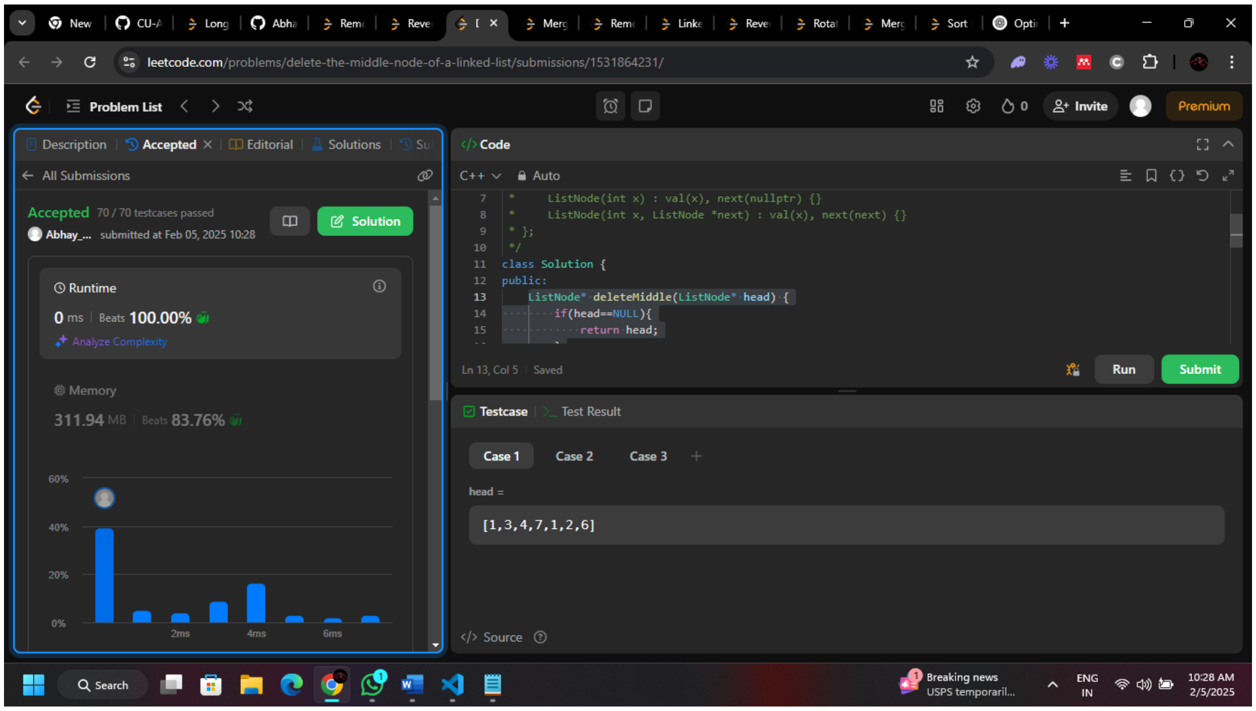
```
            next=temp->next;
            temp->next=prev;
            prev=temp;
            temp=next;
        }
        return prev;
    }
};
```

Ln 29, Col 6 | Saved

head =
[1,2,3,4,5]

3. Delete Middle Node Of A List-

```
ListNode* deleteMiddle(ListNode* head) {
        if(head==NULL){
            return head;
        }
        if(head->next==NULL){
            head=head->next;
            return head;
        }
        ListNode* fast=head;
        ListNode* slow=head;
        ListNode* prev=NULL;
        while(fast!=NULL && fast->next!=NULL){
            fast=fast->next->next;
            prev=slow;
            slow=slow->next;
        }
        prev->next=slow->next;
        slow=slow->next;
        return head; }
```



leetcode.com/problems/delete-the-middle-node-of-a-linked-list/submissions/1531864231/

Accepted 70 / 70 testcases passed
Abhay_... submitted at Feb 05, 2025 10:28

0 ms | Beats 100.00%

311.94 MB | Beats 83.76%

```
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* deleteMiddle(ListNode* head) {
        if(head==NULL){
            return head;
```

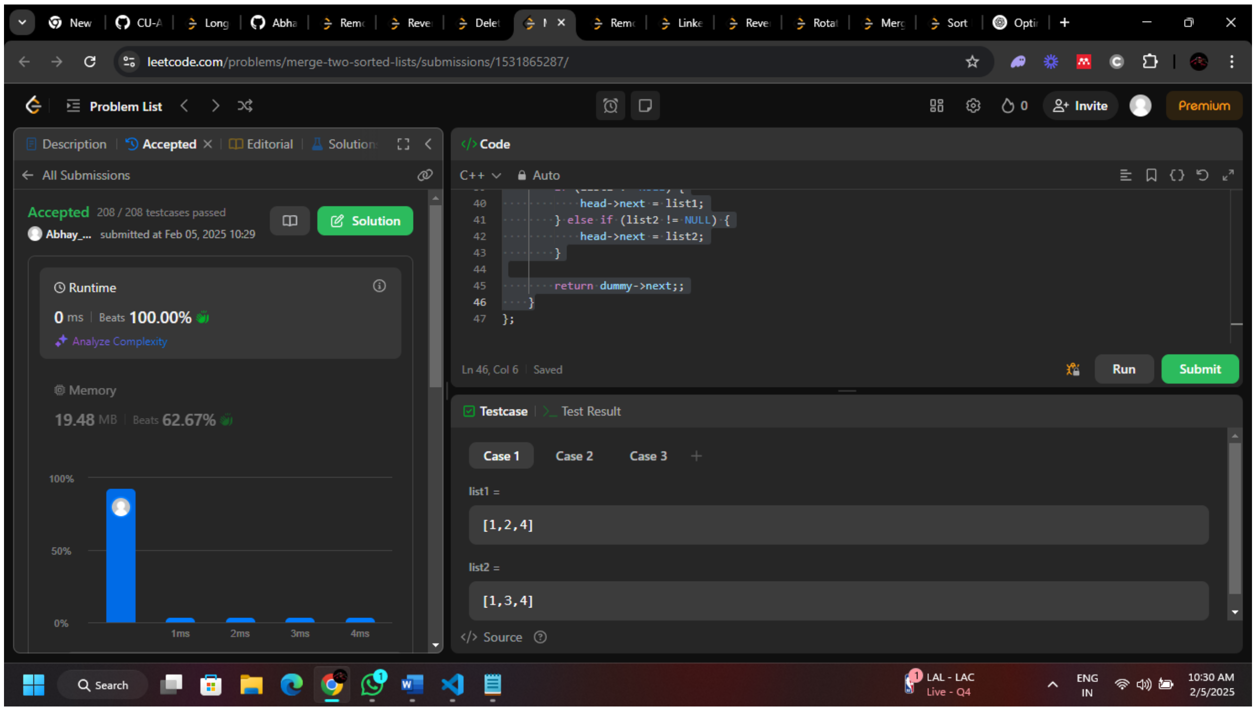
Ln 13, Col 5 Saved

head =

```
[1,3,4,7,1,2,6]
```

4. Merge Two Sorted Liked List-

```
ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        if (list1 == NULL && list2 == NULL) {
            return NULL;
        }

        if (list1 == NULL) {
            return list2;
        }
        if (list2 == NULL) {
            return list1;
        }

        ListNode* dummy = new ListNode(-1);
        ListNode* head = dummy;

        while (list1 != NULL && list2 != NULL) {
            if (list1->val <= list2->val) {
                head->next = list1;
                list1 = list1->next;
            } else {
                head->next = list2;
                list2 = list2->next;
            }
            head = head->next;
        }

        if (list1 != NULL) {
            head->next = list1;
        } else if (list2 != NULL) {
            head->next = list2;
        }

        return dummy->next;;
    }
```



leetcode.com/problems/merge-two-sorted-lists/submissions/1531865287/

Accepted 208 / 208 testcases passed

Abhay_... submitted at Feb 05, 2025 10:29

Runtime
0 ms | Beats 100.00%

Memory
19.48 MB | Beats 62.67%

```
            head->next = list1;
        } else if (list2 != NULL) {
            head->next = list2;
        }

        return dummy->next;;
    }
};
```

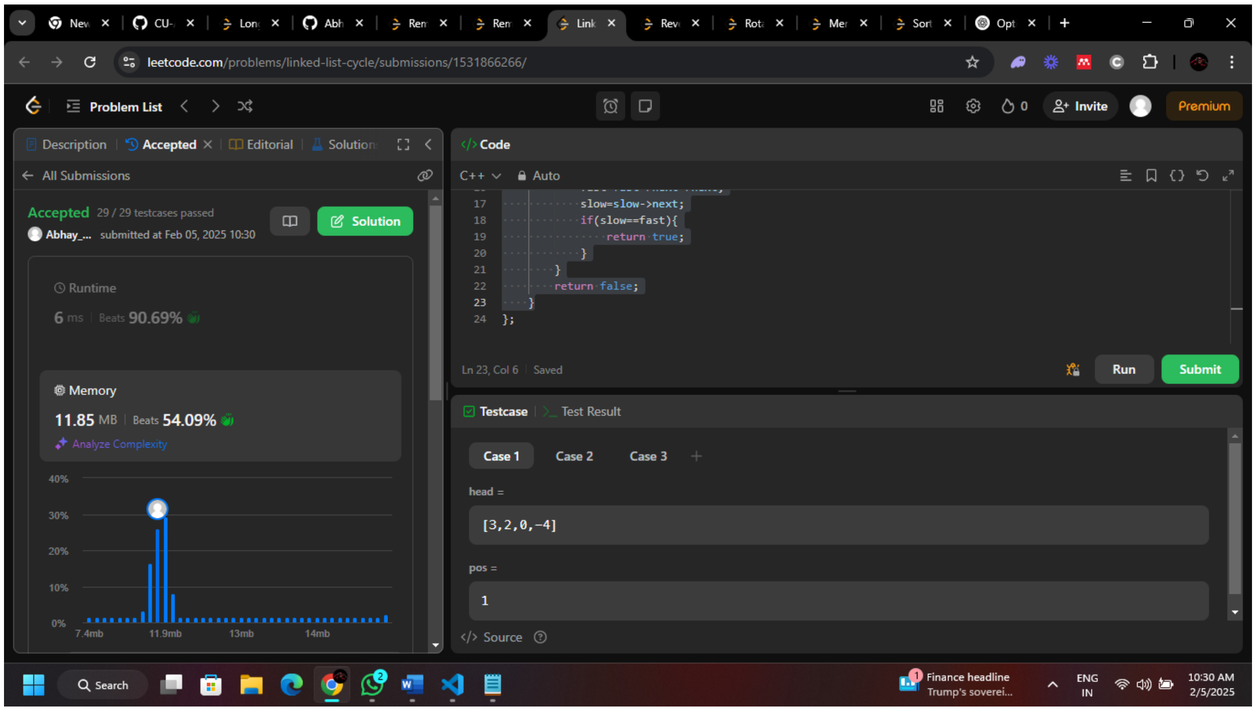
Ln 46, Col 6 | Saved

list1 =
```
[1,2,4]
```

list2 =
```
[1,3,4]
```

5. Detect A Cycle In A Linked List-

```
bool hasCycle(ListNode *head) {
        ListNode* fast=head;
        ListNode* slow=head;

        while(fast!=NULL && fast->next!=NULL){
            fast=fast->next->next;
            slow=slow->next;
            if(slow==fast){
                return true;
            }
        }
        return false;
    }
```



leetcode.com/problems/linked-list-cycle/submissions/1531866266/

Accepted 29 / 29 testcases passed
Abhay_... submitted at Feb 05, 2025 10:30

Runtime
6 ms | Beats 90.69%

Memory
11.85 MB | Beats 54.09%

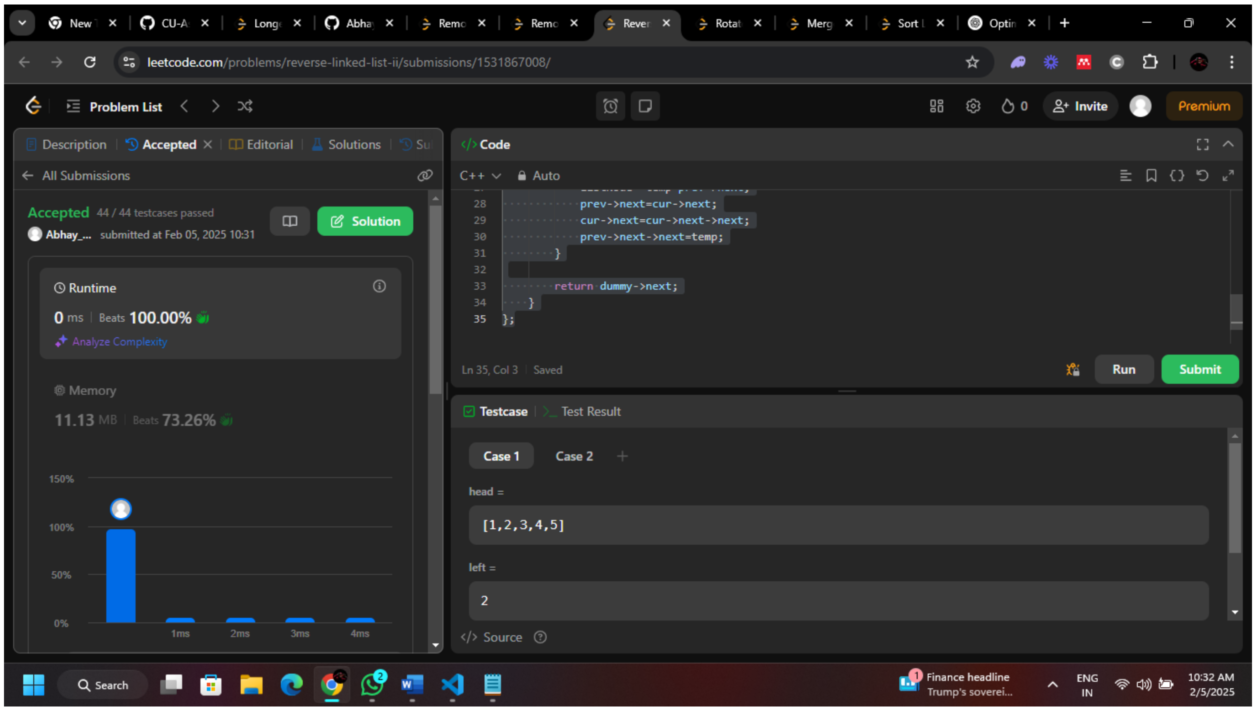
```
            slow=slow->next;
            if(slow==fast){
                return true;
            }
        }
        return false;
    }
};
```

Ln 23, Col 6 | Saved

Testcase | Test Result
Case 1 Case 2 Case 3
head =
[3,2,0,-4]
pos =
1

6. Reverse Linked List 2-

```
ListNode* reverseBetween(ListNode* head, int left, int right) {
        if(head==NULL || head->next==NULL){
            return head;
        }
        ListNode* dummy=new ListNode(-1);
        dummy->next=head;
        ListNode* prev=dummy;
        for(int i=1;i<left;i++){
            prev=prev->next;
        }
        ListNode* cur=prev->next;
        for(int i=0;i<right-left;i++){
            ListNode* temp=prev->next;
            prev->next=cur->next;
            cur->next=cur->next->next;
            prev->next->next=temp;
        }
        return dummy->next;
    }
```


leetcode.com/problems/reverse-linked-list-ii/submissions/1531867008/

Accepted 44 / 44 testcases passed
Abhay_... submitted at Feb 05, 2025 10:31

Runtime
0 ms | Beats 100.00%

Memory
11.13 MB | Beats 73.26%

```
            prev->next=cur->next;
            cur->next=cur->next->next;
            prev->next->next=temp;
        }

        return dummy->next;
    }
};
```

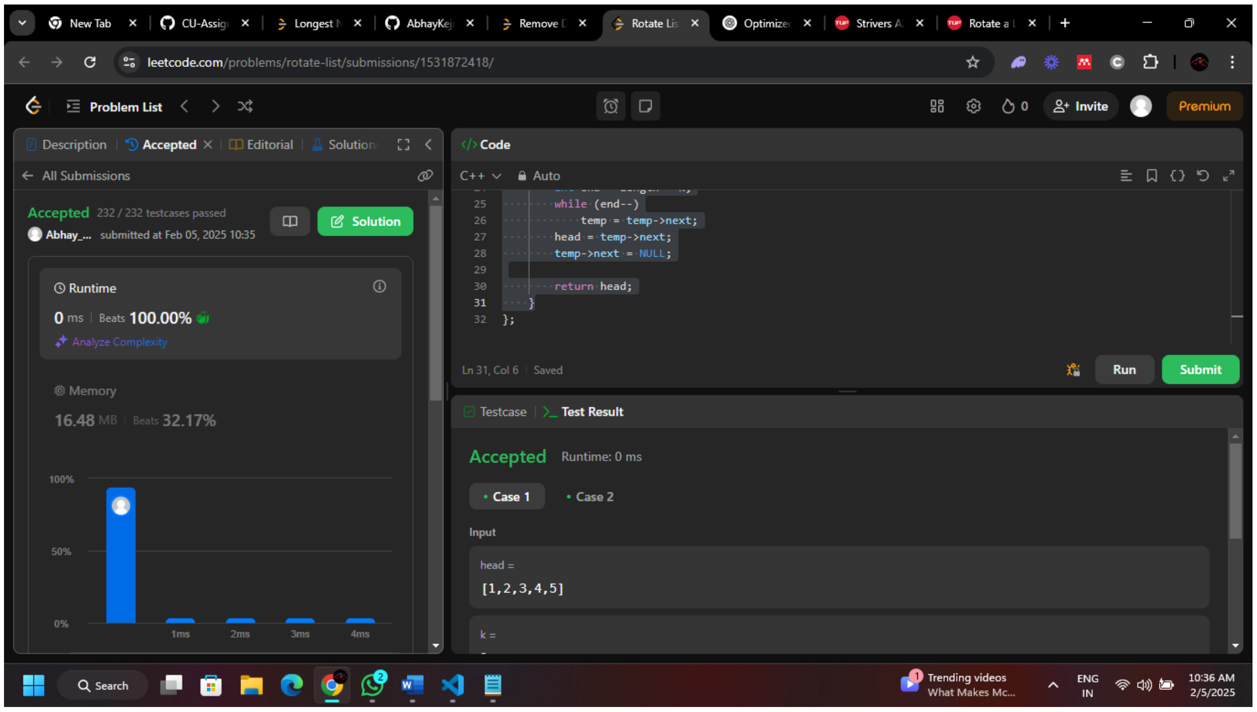
Ln 35, Col 3 Saved

head =
[1,2,3,4,5]

left =
2

7. Rotate A List-

```
ListNode* rotateRight(ListNode* head, int k) {
        if (head == NULL || head->next == NULL || k == 0)
            return head;
        ListNode* temp = head;
        int length = 1;
        while (temp->next != NULL) {
            ++length;
            temp = temp->next;
        }
        temp->next = head;
        k = k % length;
        int end = length - k;
        while (end--)
            temp = temp->next;
        head = temp->next;
        temp->next = NULL;

        return head;
    }
```



leetcode.com/problems/rotate-list/submissions/1531872418/

Accepted 232 / 232 testcases passed
Abhay_... submitted at Feb 05, 2025 10:35

Runtime
0 ms | Beats 100.00%

Memory
16.48 MB | Beats 32.17%

```
        while (end--)
            temp = temp->next;
        head = temp->next;
        temp->next = NULL;

        return head;
    }
};
```

Ln 31, Col 6 Saved

Accepted Runtime: 0 ms

Case 1 Case 2

Input

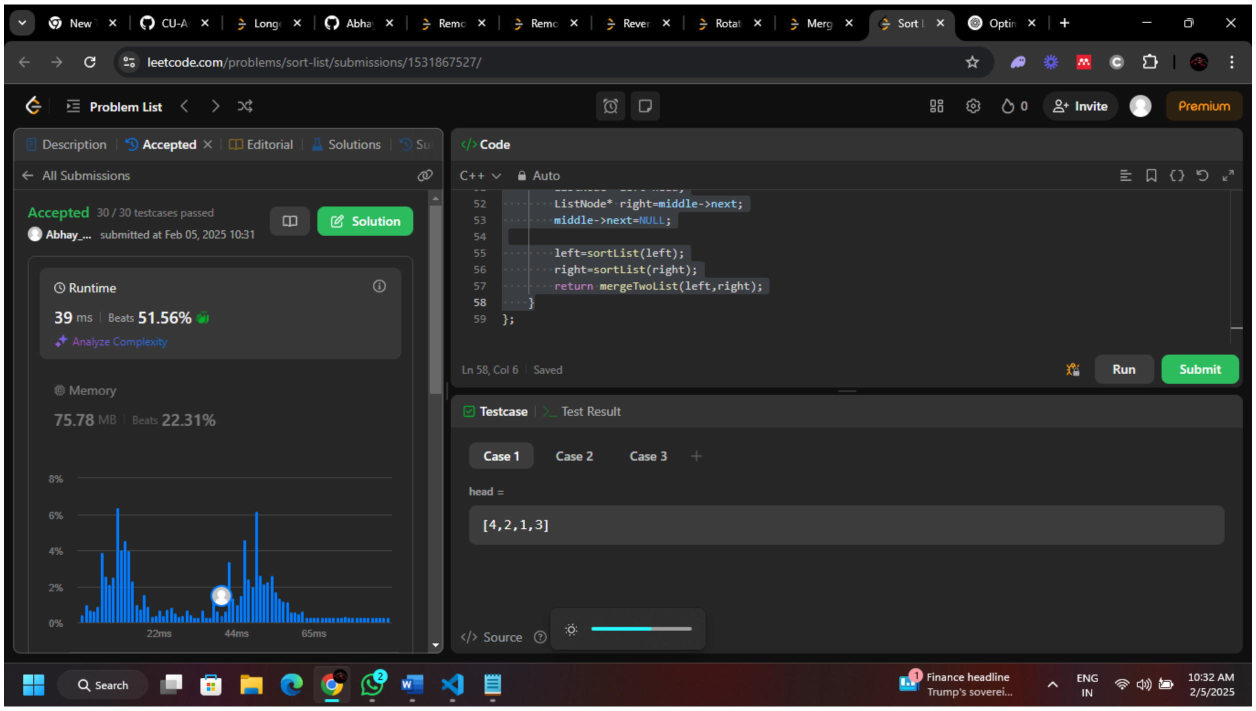
head =
[1,2,3,4,5]

k =

8. Sort List –

```
ListNode* findMiddle(ListNode* head){
        ListNode* slow=head;
        ListNode* fast=head->next;

        while(fast!=NULL && fast->next!=NULL){
            slow=slow->next;
            fast=fast->next->next;
        }
        return slow;
    }

    ListNode* mergeTwoList(ListNode* left, ListNode* right){
        ListNode* dummy=new ListNode(-1);
        ListNode* temp=dummy;

        while(left!=NULL && right!=NULL){
            if(left->val < right->val){
                temp->next=left;
                temp=left;
                left=left->next;
            }
            else{
                temp->next=right;
                temp=right;
                right=right->next;
            }
        }

        if(left)temp->next=left;
        else temp->next=right;

        return dummy->next;
    }

    ListNode* sortList(ListNode* head) {
       if (head==NULL || head->next==NULL)return head;
```

```
        ListNode* middle=findMiddle(head);
        ListNode* left=head;
        ListNode* right=middle->next;
        middle->next=NULL;

        left=sortList(left);
        right=sortList(right);
        return mergeTwoList(left,right);
    }
```



leetcode.com/problems/sort-list/submissions/1531867527/

Accepted 30 / 30 testcases passed
Abhay_... submitted at Feb 05, 2025 10:31

Runtime
39 ms | Beats 51.56%

Memory
75.78 MB | Beats 22.31%

```
        ListNode* right=middle->next;
        middle->next=NULL;

        left=sortList(left);
        right=sortList(right);
        return mergeTwoList(left,right);
    }
};
```

Ln 58, Col 6 | Saved

head =
[4,2,1,3]